

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07282023 A**(43) Date of publication of application: **27 . 10 . 95**

(51) Int. Cl.

G06F 15/163(21) Application number: **06068293**(22) Date of filing: **06 . 04 . 94**(71) Applicant: **HITACHI LTD**(72) Inventor:
**TANAKA SHIGEYA
HOTTA TAKASHI
KATSURA AKIHIRO
MORIOKA MICHIO**(54) **DATA TRANSFER AMOUNT VARIABLE
PROCESSOR AND SYSTEM USING THE SAME**

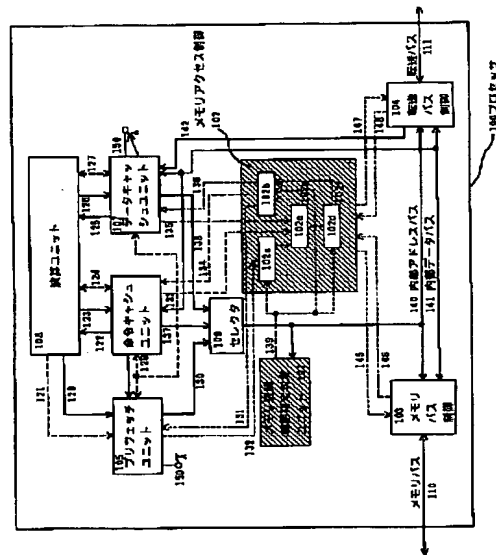
can be realized.

COPYRIGHT: (C)1995,JPO

(57) Abstract:

PURPOSE: To reduce the use rate of a communication bus and to improve the processing ability of a multiprocessor by providing a memory access control part executing the access making data transfer amounts to be different in a memory bus and a transfer bus for a processor.

CONSTITUTION: The processor 100, is provided with a judgement unit for memory space range designation 101, a memory access control 102, a memory bus control 103, a transfer bus control 104, a prefetch unit 105, an instruction cache unit 106, a data cache unit 107 and an arithmetic unit 108. It is judged whether data to be accessed is in a distributed memory corresponding to the processor 100 or in the other memory. For executing the access based on the judgement, the data transfer amount of access is changed, namely, the data transfer amount to the transfer bus 111 is reduced compared to that to the memory bus 110. Thus, the use time of the transfer bus 111 per one time becomes short, the use rate of the transfer bus 111 is reduced and a high speed processing



P 2252

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-282023

(43)公開日 平成7年(1995)10月27日

(51)Int.Cl.⁶

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/163

G 0 6 F 15/ 16

3 2 0 M

審査請求 未請求 請求項の数14 O L (全 23 頁)

(21)出願番号

特願平6-68293

(22)出願日

平成6年(1994)4月6日

(71)出願人

000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者

田中 成弥

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72)発明者

堀田 多加志

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72)発明者

桂 晃洋

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(74)代理人

弁理士 小川 勝男

最終頁に続く

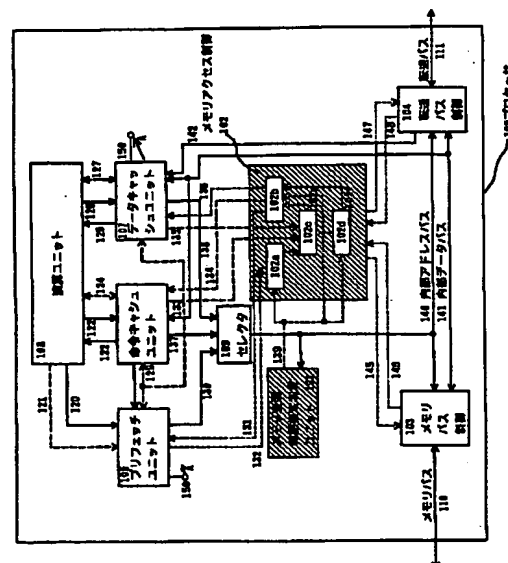
(54)【発明の名称】 データ転送量可変プロセッサ及びそれをを用いたシステム

(57)【要約】

【目的】本発明の目的は、共有分散メモリ型のマルチプロセッサシステムにおいてデータの転送量を可変にすることにある。

【構成】データを分散して保持する分散メモリをメモリバスを介してアクセスするとともに少なくとも1つの他の分散メモリを転送バスを介してアクセスしてデータを処理するプロセッサであって、上記プロセッサは、上記メモリバスと上記転送バスとでデータの転送量を異ならせてアクセスするメモリアクセス制御部を有することを特徴とするデータ転送量可変プロセッサ。

図 1



ス制御部を有することを特徴とするデータ転送量可変プロセッサ。

【請求項8】データを分散して保持する分散メモリをメモリバスを介してアクセスするとともに少なくとも1つの他の分散メモリを転送バスを介してアクセスしてデータを処理するプロセッサであって、
上記プロセッサは、データを処理する演算部と上記分散メモリに保持されているデータの一部を保持するキャッシュメモリ部と上記演算部の処理に応じて上記キャッシュメモリ部へデータを書き込むプリフェッチ部とを有し、上記プリフェッチ部によって上記キャッシュメモリ部に書き込まれたデータが上記演算部のアクセスするデータでなければ、上記プリフェッチ部は上記演算部の次の処理に応じて上記キャッシュメモリ部へ次のデータを書き込むために、上記次の書き込むデータが上記分散メモリに保持されているか上記他の分散メモリに保持されているかを判定し、上記分散メモリに保持されているなら上記メモリバスを介して第1の転送量でアクセスし、上記他の分散メモリに保持されているなら上記転送バスを介して第2の転送量でアクセスするメモリアクセス制御部を有することを特徴とするデータ転送量可変プロセッサ。

【請求項9】データを分散して保持する分散メモリをメモリバスを介してアクセスするとともに少なくとも1つの他の分散メモリを転送バスを介してアクセスしてデータを処理するプロセッサであって、
上記プロセッサは、データを処理する演算部と、上記分散メモリに保持されているデータの一部を保持し、上記演算部のアクセスするデータを保持しているかを判定するキャッシュメモリ部と、上記演算部の処理に応じて上記キャッシュメモリ部へデータを書き込み、上記キャッシュメモリ部に書き込まれたデータが上記演算部のアクセスするデータでなければ、上記演算部の次の処理に応じて上記キャッシュメモリ部へ次のデータを書き込むプリフェッチ部と、全ての分散メモリのアドレス空間情報を保持し、上記次の書き込むデータのアドレスを上記アドレス空間情報に基づいて上記次の書き込むデータが上記分散メモリに保持されているか上記他の分散メモリに保持されているかを判定するメモリ空間範囲判定部と、上記メモリ空間範囲判定部の判定によって、上記分散メモリに保持されているなら上記メモリバスを介して第1の転送量でアクセスし、上記他の分散メモリに保持されているなら上記転送バスを介して第2の転送量でアクセスするメモリアクセス制御部を有することを特徴とするデータ転送量可変プロセッサ。

【請求項10】請求項1乃至9のいずれか1項に記載の上記データ転送量可変プロセッサと、上記データ転送量可変プロセッサとメモリバスによって接続され、全アドレス領域の相異なるアドレス領域を有する分散メモリとからなる命令処理部を複数有し、上記それぞれの命令処

理部は転送バスによって接続されていることを特徴とする分散処理システム。

【請求項11】請求項10に記載の分散処理システムを1つのクラスタとし、上記クラスタを複数有し、上記クラスタのそれぞれはクラスタ間通信制御部と通信ネットワークによって接続され、上記クラスタ間の情報転送量は、上記分散処理システム内のデータ転送量とは異なることを特徴とする情報処理システム。

【請求項12】請求項11において、

10 上記クラスタ間通信制御部は、上記通信ネットワークを介して他のクラスタに対して転送バスプロトコルとしてアドレス、データ、プロトコル制御信号及びデータ転送量を設定、送信することを特徴とする情報処理システム。

【請求項13】請求項12において、

上記データ転送量がゼロのときには、上記ゼロにしたアドレス空間へのアクセスを停止することを特徴とする情報処理システム。

【請求項14】請求項4から13のいずれか1項において、

20 上記キャッシュメモリはブロックサイズをn個のサブブロックに分割され、データ転送量は(0からn個)×ブロックサイズであることを特徴とする情報処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、データ処理装置に係り、特に共有分散メモリにおけるデータ転送量可変プロセッサに関する。

【0002】

30 【従来の技術】従来の共有分散メモリを有するのマルチプロセッサシステムとして、特開昭56-155465号、特開平1-134656号、特開平2-244253号、特開平4-326453号に記載のように、複数のプロセッサと共有メモリ装置とを通信(転送)バスで接続し、物理アドレス空間の全部、又は一部を各プロセッサからアクセス出来る構成をとる。

40 【0003】これらの公知技術では、プロセッサ又はプロセッサと分散メモリからなる命令処理装置が、アクセスすべきデータが自己の分散メモリのメモリ空間へのアクセスか、他の分散メモリのメモリ空間へのアクセスかを判定し、自己の分散メモリ空間内であれば自己の分散メモリ装置にアクセス要求を、自己の分散メモリ空間内でなければ通信バスを通して他の分散メモリ装置にアクセス要求を出力する。

【0004】

50 【発明が解決しようとする課題】上述の従来技術に示された共有分散メモリを有するのマルチプロセッサシステムは、データを分散しない1つのメモリをアクセスするものに比べて通信バス要求回数の低いマルチプロセッサシステムが提供できる。

【0005】しかし、一回の通信バス要求に対するデータ又は命令（以下、命令もデータとして扱う）の転送量については何も記載されていない。

【0006】さらに、プロセッサにキャッシュメモリを内蔵するものが一般的になってきている。このようなプロセッサ又はシステムにおいて、キャッシュヒットしなかった場合のアクセス、今後使われるであろうデータに対して先行してメモリアccessを起す先行プリフェッチとの関係についても記載されていない。

【0007】本発明の第1の目的は、分散共有メモリ型のマルチプロセッサシステムにおいて、通信バスの占有する時間を短くすべく、バス要求回数の低減と1回当りの転送量の最適化をおこない、通信バスの使用率を下げ、マルチプロセッサ全体の処理能力を向上させることである。

【0008】さらに、本発明の第2の目的は、分散共有メモリ型のマルチプロセッサシステムにおいて、通信バスのバスネックを起こさないようにしながら可能な限り先行プリフェッチをおこない、マルチプロセッサ全体の処理能力を向上させるシステムを提供することにある。

【0009】

【課題を解決するための手段】上記第1の目的を達成するために、データを分散して保持する分散メモリをメモリバスを介してアクセスするとともに少なくとも1つの他の分散メモリを転送バスを介してアクセスしてデータを処理するプロセッサであって、上記プロセッサは、上記メモリバスと上記転送バスとでデータの転送量を異ならせてアクセスするメモリアccess制御部を有することを特徴とする。

【0010】さらに、上記第2の目的を達成するためには、データを分散して保持する分散メモリをメモリバスを介してアクセスするとともに少なくとも1つの他の分散メモリを転送バスを介してアクセスしてデータを処理するプロセッサであって、上記プロセッサは、データを処理する演算部と上記分散メモリに保持されているデータの一部を保持するキャッシュメモリ部と上記演算部の処理に応じて上記キャッシュメモリ部へデータを書き込むプリフェッチ部とを有し、上記プリフェッチ部によって上記キャッシュメモリ部に書き込まれたデータが上記演算部のアクセスするデータでなければ、上記プリフェッチ部は上記演算部の次の処理に応じて上記キャッシュメモリ部へ次のデータを書き込むために、上記次の書き込むデータが上記分散メモリに保持されているか上記他の分散メモリに保持されているかを判定し、上記メモリバスと上記転送バスとでデータの転送量を異ならせてアクセスするメモリアccess制御部を有することを特徴とする。

【0011】

【作用】上記第1の特徴によると、アクセスすべきデータがプロセッサに対応した分散メモリにあるか他の分散

メモリにあるかを判定し、その判定に基づいてアクセスするのに、アクセスのデータ転送量を変えること、つまり、転送バスへのデータ転送量をメモリバスへのデータ転送量に比べて小さくする。これによって、転送バスの1回当りの使用時間は短くなり、その結果転送バスの使用率を下げ、プロセッサ又はシステム全体として高速な処理が達成される。

【0012】さらに、上記第2の目的は、プロセッサは、主記憶のデータのコピーを持つキャッシュメモリと、プロセッサの処理に応じて今後使われるであろうデータに対して先行してメモリアccessを行う先行プリフェッチ部とを有し、プリフェッチ要求されたプリフェッチデータが存在しないときは、要求されたアドレスがプロセッサに対応した分散メモリのメモリ空間内か否かを判定し、この判定結果によるアクセスすべきメモリ空間に応じたデータ転送量、つまり、一般には転送バスへのデータ転送量をメモリバスへのデータ転送量に比べて小さくして、アクセスする。これによって、転送バスの1回当りの使用時間は短くなり、その結果転送バスの使用率を下げるように働く。

【0013】また、当該命令処理手段外のメモリ装置への先行プリフェッチでのメモリアccessを行わないようにする。これによって、転送バスへのアクセス回数が少なくなり、その結果転送バスの使用率を下げるように働く。

【0014】このように、転送バスの使用率を下げることにより、マルチプロセッサの台数を増やせるか又はバス待ち時間が減りシステム性能が向上する。

【0015】

【実施例】図1に本実施例のプロセッサのブロック図を示す。100はプロセッサ、101はメモリ空間範囲指定用判定ユニット、102はメモリアccess制御、103はメモリバス制御、104は転送バス制御、105はプリフェッチユニット、106は命令キャッシュユニット、107はデータキャッシュユニット、108は演算ユニットで構成する。また、プロセッサから外部に接続されるバスは、メモリバス110、転送バス111である。

【0016】図2に図1で示したプロセッサを使用した共有分散メモリ型のマルチプロセッサの構成を示す。プロセッサ201と分散メモリー1（205）とそれを接続するメモリバス215で1つの命令処理装置を構成し、それらを複数個持ち各プロセッサ間を転送バス213で接続する。201から204は図1で示したプロセッサ、205から208は分散メモリ、209はI/Oコントローラ、210～212はディスプレイ、プリンタ、ディスク等のI/Oデバイスで構成する。プロセッサ100の転送バス111が図2の転送バス213に接続され、図1のプロセッサ100のメモリバス110がメモリバス215に接続する構成であり、プロセッ

サ202から204についても同様である。

【0017】図2の分散メモリのアドレス空間を図3に示す。本実施例のシステムは物理アドレス4Gバイトを実装しており、メモリマップI/Oの構成をとる。このため各プロセッサ(210~204)から見えるアドレス空間は4Gバイトであり、アドレスX' F0000000からX' FFFFFFFFまでがI/O空間である。図2の分散メモリ205から208は図3に示す各分散メモリの実装アドレス空間を割当て、それぞれの分散メモリ上で実装する。

【0018】さて、始めに図1から図3を用いてマルチプロセッサシステムの基本動作を述べ、続いてプロセッサ内部の詳細な動作を説明する。

【0019】図2において、プロセッサ1を基準に考えると、プロセッサ1がアドレスX' 00000100なるアドレスをアクセスする場合、プロセッサ1内部の制御により、プロセッサ1が直接接続している分散メモリ1へのアクセスであることを認識し、プロセッサ1はメモリバス215を起動して分散メモリ1(205)へのアクセスを行う。一方、プロセッサ1がアドレスX' 4F000000なるアドレスをアクセスする場合、プロセッサ1内部の制御により、プロセッサ1が転送バスへのアクセスであることを認識し、プロセッサ1は転送バス213を起動する。プロセッサ1から4(201から204)は転送バスを常に監視しており、このケースでは、プロセッサ2から4が各プロセッサに直接接続している分散メモリへのアクセスか否かを調べる。本例では、プロセッサ2が直接接続している分散メモリ2へのアクセスであることを認識しデータを受け取る。プロセッサ2は転送バスから受けとったデータより、分散メモリ2へのアクセスをするための準備としてプロセッサ2の内部で転送バスとメモリバス216を接続する。このような一連の操作により、プロセッサ1(201)、転送バス(213)、プロセッサ2(202)、メモリバス216、分散メモリ2(206)と接続されアクセスすることができる。さらに、プロセッサ1がアドレスX' F1000000なるアドレスをアクセスする場合、プロセッサ1内部の制御により、プロセッサ1が転送バスへのアクセスであることを認識し、プロセッサ1は転送バス213を起動する。本実施例では、I/Oコントローラ209が転送バス213を監視してI/Oエリアへのアクセスであることを認識しデータを受け取る。その後、I/Oコントローラ209は、各I/Oデバイス210から212へアクセスするためI/Oコントローラ209内で転送バスとI/Oバスを接続する。このような一連の操作により、プロセッサ1(201)、転送バス(213)、I/Oコントローラ209、I/Oバス(214)、I/Oデバイス(210~212)と接続されアクセスすることができる。

【0020】本実施例のマルチプロセッサは、各プロセッサ内部にキャッシュメモリが内蔵されており、主記憶のデータと常に一致させる一致化手段として各プロセッサは転送バスを監視するスヌープキャッシュ構成をとる。本実施例では、あるプロセッサが他のプロセッサ内のキャッシュメモリに対してデータの一致性を確認するとき、CCC(キャッシュコヒーレントチェック)要求を出力する。CCC要求は、チェックすべきアドレスを転送バスを介してブロードキャストされる。本実施例では、プロセッサ内部のアドレスアレが2ポートになっているため各プロセッサ自身の実行している内部動作と独立にチェックできる構成をとる。さらに、プロセッサ1が、キャッシュミスして、プロセッサ1から分散メモリ1へアクセスしている場合にも、他のプロセッサからのCCC要求が独立に実行できる。

【0021】続いて、プロセッサ内部の本発明の特徴である部分を中心に詳細な動作について述べる。

【0022】図1に示すプロセッサ100の基本的な命令の動作を図4、図5、図6で説明する。図4に演算ユニット108の詳細な構成を示す。演算ユニット108は、レジスタ1301、演算器1302、命令バッファレジスタ1303、命令デコーダ1304、プログラムカウンタ1305、分岐ADDER1306、ロード命令用バイトアライナ1308、ストア命令用バイトアライナ1307、割込み処理用アドレスジェネレータ1309、パイプラインを実行するラッチ1310a~k、パイプライン制御1312、条件分岐nottaken用アドレス生成回路1313で構成される。

【0023】図5に命令キャッシュユニット106の構成を示す。1201は論理アドレスラッチ、1202は命令用TLB、1203は命令キャッシュ用アドレスアレ、1204はアドレス格納メモリ、1205はデータ格納バッファ及びプリフェッチバッファ、1206は命令オペランド格納メモリ、1207は命令キャッシュ制御ユニット、1210はTLBのVPN(バーチャルページ)比較器、1211はTLBのプロテクション情報比較器、1212はアドレスアレ用の比較器、1213はセレクタで構成する。TLB、命令キャッシュはダイレクトマップ方式をとり、命令TLBは128エントリ、命令キャッシュの容量は1Kエントリ32Kバイトである。

【0024】図6にデータキャッシュユニット106の構成を示す。1001は論理アドレスラッチ、1002はデータ用TLB、1003はデータキャッシュ用アドレスアレ、1004はアドレス格納バッファ、1005はデータ格納バッファ、1006はデータオペランド格納メモリ、1007はデータキャッシュ制御ユニット、1010はTLBのVPN(バーチャルページ)比較器、1011はTLBのプロテクション情報比較器、1012はアドレスアレ用の比較器、1013はセレクタ、10

14はCCC(キャッシュコヒーレンスチェック)のための比較器で構成する。TLB、命令キャッシュはダイレクトマップ方式をとり、データTLBは128エントリ、データキャッシュの容量は1Kエントリ32Kバイトである。

【0025】パイプライン動作は図7に示すように6ステージで実行される。命令セットは一般のRISCで用いられるレジスタ間の演算命令、分岐命令、ロード命令、ストア命令を持つ。これらの命令の基本動作は、本

発明と直接関係無いので、メモリアクセスと関係のある命令フェッチとデータフェッチの部分だけを述べる。

【0026】IFステージにおいて、命令キャッシュユニット106では、演算ユニット108からの命令アドレス123を図5に示すように論理アドレスラッチ1201にセットし、命令フェッチリクエスト124により命令キャッシュユニット106に起動がかかる。論理アドレスは2-4ビットのブロック内アドレスと5-14ビットのブロックアドレスと12-31ビットのページアドレスに分けられる。通常の動作は、ページアドレスがTLBにより論理アドレスから物理アドレスへ変換される。12-31ビットのページアドレスは、12-18ビットで命令TLB内のエントリを選択し、読みだされたアドレス1202aと19-31ビットのアドレスを比較器1210を介して比較され、一致したときTLBヒットとなる。更に、選ばれたエントリに対してプロテクション情報1202cを読みだし管理用レジスタと比較器1211を通してチェックする。アドレスアレ1203は論理アドレスの5-14ビットでアドレスアレ内の1Kエントリのうちの1つを選択する。読みだされた物理アドレスページ(PPN)1203aと命令TLBから読みだされた物理アドレスページ(PPN)1202bは比較器1212で比較され一致したときキャッシュヒットとなる。

【0027】なお、命令キャッシュ用アドレスアレは8バイト単位の処理ができるように4つのサブブロック有効ビット1203bを格納する。図8(a)に1エントリ当たりの1203bの内部情報を示す。更に、図8(b)にBV1から4までの組合せを示す。同一ブロック内の連続するサブブロックのみ許される制約を設ける。一方、命令オペランド格納メモリ1206は論理アドレスの5-14ビットで命令オペランド格納メモリ内の1Kエントリ内の1つを選択する。選択された32バイトの命令オペランドを、更に、ブロック内アドレス2-5ビットで8分の1に選択すると、4バイトの命令がオペランドバス122を通して演算器ユニット108内の命令バッファレジスタ1303へ送られる。

【0028】ところで、ロード命令は、演算ユニット108からのデータアドレス126を図6に示すように論理アドレスラッチ1001にセットし、データフェッチリクエスト127によりデータキャッシュユニット10

7に起動がかかる。論理アドレスは2-4ビットのブロック内アドレスと5-14ビットのブロックアドレスと12-31ビットのページアドレスに分けられる。ページアドレスがTLBにより論理アドレスから物理アドレスへ変換される。12-31ビットのページアドレスは、12-18ビットでデータTLB内のエントリを選択し、読みだされたアドレス1002aと19-31ビットのアドレスを比較器1010を介して比較され、一致したときTLBヒットとなる。更に、選ばれたエントリに対してプロテクション情報1002cを読みだし管理用レジスタと比較器1011を通してチェックする。アドレスアレ1003は論理アドレスの5-14ビットでアドレスアレ内の1Kエントリのうちの1つを選択する。読みだされた物理アドレスページ(PPN)1003aとデータTLBから読みだされた物理アドレスページ(PPN)1002bは比較器1012で比較され一致したときキャッシュヒットとなる。

【0029】なお、データキャッシュ用アドレスアレは8バイト単位の処理ができるように4つのサブブロック有効ビット1003bを格納する。サブブロックは命令キャッシュと同様の構成をとる。一方、データ格納メモリ1006は論理アドレスの5-14ビットでデータ格納メモリ内の1Kエントリ内の1つを選択する。選択された32バイトのデータを、更に、ブロック内アドレス2-5ビットで8分の1に選択すると、4バイトの命令がデータバス125を通して演算器ユニット108内へ送られる。

【0030】ストア命令の動作は、Eステージで演算ユニット108からのデータアドレス126が論理アドレスラッチ1001にセットされ、データリクエスト124(ストア要求)によりデータキャッシュユニットに起動がかかる。Tステージで演算ユニット108よりデータバス125を通して、書き込むべきデータがデータキャッシュユニット107に送られ、データ格納バッファ1005に格納される。比較器1010、1011、1012の結果、TLBヒット、プロテクションヒット、キャッシュヒットであれば、ストア命令のWステージで、データ格納メモリ1006へ書き込まれて命令は完了する。

【0031】本実施例で示す通り、プロセッサ内部の演算ユニットからのメモリアクセス要求に対し、内蔵キャッシュメモリにデータが存在している場合、メモリアクセス要求に対応したデータを供給することが出来る。

【0032】次に、図1に戻って、命令キャッシュユニット106、データキャッシュユニット107からチップ外部へのメモリアクセスについて図9、図10、図11、図12により述べる。

【0033】図9はメモリ空間範囲指定用判定ユニット101の構成で、分散共有アドレス中の自己メモリ空間下限値格納レジスタ401、分散用アドレス中の自己メ

10

20

30

40

50

メモリ空間上限値格納レジスタ402, 内部アドレスバス140と下限値レジスタ401を比較する比較回路403, 内部アドレスバス140と上限値レジスタ402を比較する比較回路404, 自己メモリ空間内に内部アドレスが入っているか否かを判定する判定回路より構成される。

【0034】図10はメモリアクセス制御102の構成で、マスク回路102a, 優先判定回路102c, バス起動回路およびバスアービタ102d, プリフェッチインタフェース制御1901, 命令キャッシュユニットインタフェース制御1902, データキャッシュユニットインタフェース制御1903より構成される。

【0035】図11はメモリバス制御103で、メモリ制御回路501, データ用バッファ回路502, アドレス用バッファ回路503, スイッチ504, 505, 506から構成される。

【0036】図12は転送バス制御104で、転送バス制御回路901, データ用バッファ回路902, 903, アドレス用バッファ回路904, 905, スイッチ906~910より構成される。

【0037】続いて、命令キャッシュユニット106から命令リクエスト133がアサートされたときの動作を示す。

【0038】命令キャッシュユニット106内のアドレス比較器1212の比較の結果キャッシュミスになると命令キャッシュ制御ユニット1207は演算ユニット108に対してパイプライン処理を待つように指示(図中では示していない)し、信号133を通して、メモリアクセス制御102に命令フェッチ要求を出す。アドレス格納バッファ1204は物理アドレスを生成し、命令アドレス139としてセレクタ109を介して、内部アドレスバス140に出力する。メモリ空間指定用判定ユニットは、常に内部アドレスバス140を監視している。

【0039】メモリ空間範囲指定用判定ユニット101の動作を説明する。各プロセッサは、システム立ち上げのとき、下限値格納レジスタ401, 上限値格納レジスタ402にソフトウェア上で、設定すべきデータを406を介してセットする。読みだし用407と書き込み用406のバスは、演算器ユニット内のレジスタ1301とコントロールレジスタ用の転送命令によりデータ転送ができる(図中省略)。各プロセッサは直接メモリバスを介してアクセス可能な各分散メモリの下限, 上限を設定する。本実施例の場合、図3より、図13(a)のような値を設定する。

【0040】比較回路403は内部アドレスバス140が下限レジスタ401以上であることを検出するように動作する。また、比較回路404は内部アドレスバス140が上限レジスタ401以下であることを検出するように動作する。判定回路405は、比較回路403の出力410と比較回路404の出力420を入力し、直接接

続されている分散メモリの空間内であることを判定する。判定された信号139は、メモリアクセス制御102へ送られる。判定回路405は信号線410と420の論理積により実現できる。

【0041】命令キャッシュからのリクエスト133は優先判定回路102cに入る。いくつかのリクエストが同時に入ってきたときに102cはリクエストの順序付けを行う。優先判定の結果、命令キャッシュからのリクエストが選ばれれば信号1907がアサートされる。この信号はプロセッサ内部リクエストが外部に対して選択されたリクエストである。バス起動回路及びバスアービタ102dは外部からのリクエスト146, 148と共にどのリクエストが優先かを判定して優先度の最も高いものを受付ける。リクエスト1907が受け付けられると、バス起動回路及びバスアービタ102dはメモリバス110, 転送バス111のどちらに起動をかけるかを信号139より判定する。もし、139がアサートされていれば、信号145を通してメモリバス制御103へブロック転送(32バイト)要求を出力する。もし、139がネゲートされていれば、信号147を通して転送バス制御104へサブブロック転送(8バイト)要求を出力する。メモリバスと転送バスで転送量が異なる理由として、メモリバスはプロセッサが直接アクセスできる分散メモリへの転送手段であるが、転送バスは、他のプロセッサの分散メモリへのアクセスするための転送手段であり、4つのプロセッサが共有しており、転送量を減らすことで転送バスの使用率を下げるためである。

【0042】さらに、優先判定回路102cは信号1905をアサートさせ命令キャッシュユニットインタフェース制御1902にリクエストが受け付けられたことを知らせる。命令キャッシュユニットインタフェース制御1902は、信号1905とメモリ空間範囲指定用判定ユニット101からの信号139によって命令キャッシュユニット106へ応答する信号を作成する。134aは応答信号であり、リクエストが受け付けられたときにアサートする。134bは付属情報でありブロック転送(32バイト)かサブブロック転送(8バイト)で受け付けられたかを応答するためのものである。この134a, 134bを受け取った命令キャッシュユニット106は、リクエストが受け付けられたこと、そのリクエストに対するブロックサイズがブロック転送かサブブロック転送である事を知り、それに合わせてデータがくるのを待つ。

【0043】命令キャッシュメモリからのリクエストがプロセッサに直接接続されているメモリへのアクセスである場合の動作について述べる。メモリバス制御103は、メモリバスリクエスト145により起動がかかる。メモリバス制御103の動作は分散メモリへのリードライトによりスイッチの動作を信号510, 516, 517, 518を使って制御する。

10

20

30

40

50

【0044】続いて、命令キャッシュメモリからのリクエストがプロセッサに直接接続されたメモリへのアクセスでない別の分散メモリへのアクセス（つまり転送バスへのアクセス）の場合の動作について述べる。転送バス制御104は、転送バスリクエスト147により起動がかかる。メモリバス制御104の動作は、プロセッサと直接接続されていない分散メモリ、I/Oへのリード、ライトアクセスをスイッチの動作を信号930、931、932、933、934、935を使って制御する。

【0045】なお、転送バス制御は、リクエストにより起動していないとき、111と142を接続するようにスイッチ909、910が制御されており、転送バス111を内部キャッシュコヒーレントチェック用アドレスを通して監視できる。

【0046】以上、メモリバスまたは転送バスより読みだされたデータは、内部バス141を通して命令キャッシュユニット106へ転送される。同時に、制御信号は146または148よりメモリアクセス制御102へ報告され、これを受けて102f、134を伝わり命令キャッシュユニット106に知らされる。

【0047】つぎに、データキャッシュユニット107からデータリクエスト135がアサートされたときの動作を示す。データキャッシュからのリクエスト135は優先判定回路102cに入る。優先判定の結果、データキャッシュからのリクエストが選ばれ、図10の信号1907がアサートされる。バス起動回路102dがリクエスト1907を受け付けると、バス起動回路はメモリバス、転送バスのどちらかに起動をかけるかを信号139より判定する。もし、139がアサートされていれば、信号145を通してメモリバス制御103へブロック転送（32バイト）要求を出力する。もし、139がネゲートされていれば、信号147を通して転送バス制御104へサブブロック転送（8バイト）要求を出力する。

【0048】さらに、優先判定回路102cは信号1906をアサートさせデータキャッシュユニットインタフェース制御1903にリクエストが受け付けられたことを知らせる。命令キャッシュユニットインタフェース制御1903は、信号1906とメモリ空間範囲指定用判定ユニット101からの信号139によってデータキャッシュユニット107へ応答する信号を作成する。136aは応答信号であり、リクエストが受け付けられたときにアサートする。136bは付属情報でありブロック転送（32バイト）かサブブロック転送（8バイト）で受け付けられたかを応答するためのものである。この136a、136bを受け取ったデータキャッシュユニット107は、リクエストが受け付けられたこと、そのリクエストに対するブロックサイズがブロック転送かサブブロック転送である事を知り、それに合わせてデータが

くるのを待つ。

【0049】データキャッシュメモリからのリクエストがプロセッサに直接接続されているメモリへのアクセスである場合の動作について述べる。メモリバス制御103は、メモリバスリクエスト145により起動がかかる。メモリバス制御103の動作は分散メモリへのリードライトによりスイッチの動作を信号510、516、517、518を使って制御する。

【0050】命令キャッシュメモリからのリクエストがプロセッサに直接接続されたメモリへのアクセスでない別の分散メモリへのアクセス（つまり転送バスへのアクセス）の場合の動作について述べる。転送バス制御104は、転送バスリクエスト147により起動がかかる。メモリバス制御104の動作は、プロセッサと直接接続されていない分散メモリ、I/Oへのリード、ライトアクセスをスイッチの動作を信号、930、931、932、933、934、935を使って制御する。

【0051】メモリバスまたは転送バスより読みだされたデータは、内部バス141を通してデータキャッシュユニット107へ転送される。同時に、制御信号は146または148よりメモリアクセス制御102へ報告され、これを受けて102f、136を伝わりデータキャッシュユニット107に知らされる。

【0052】プロセッサ内部から外部メモリのアクセスへのリクエストの最後として、前もって予測した先行フェッチを行いメモリアクセスすべきデータがあらかじめキャッシュメモリに存在しているようにしたプリフェッチ方式を述べる。なお、プリフェッチ方式は、あくまでも予測して先行プリフェッチするものであるので実際にフェッチしなくても良いという特徴がある。本実施例では、命令のプリフェッチとデータのプリフェッチを行う。図14(a)、(b)に概念を示す。命令のプリフェッチ(a)は、命令キャッシュユニット106がキャッシュミスしたとき、その次のブロックをプリフェッチするものでプリフェッチユニット内の演算器で次のブロックのアドレス計算し、メモリアクセスにより得た情報はプリフェッチバッファに格納される。

【0053】データのプリフェッチ(b)はロード、ストア命令のポストインクリメントと、プリデクリメント機能を利用する。

【0054】ロード命令、ストア命令の命令機能を図15に示す。

【0055】命令「LOAD」は、ディスプレイメント+汎用レジスタ(b)がアドレスとなり、そのアドレスの指す内容は汎用レジスタ(t)に格納する。命令「STORE」は、ディスプレイメント+汎用レジスタ(b)がアドレスとなり、汎用レジスタ(r)の内容がアドレスで指されたメモリへ書き込まれる。

【0056】さらに、各命令共にアドレッシングによりさらに2つのバリエーションを持つ。「---PI」

は、アドレス計算せず汎用レジスタ (b) がアドレスとなると共に、同時にディスプレースメント+汎用レジスタ (b) を計算して、その値を汎用レジスタ (b) に格納する。このアドレッシングモードはいわゆるポストインクリメント機能を有する。「---PD」は、ディスプレースメント+汎用レジスタ (b) がアドレスと共に、同時にディスプレースメント+汎用レジスタ (b) を計算して、その値を汎用レジスタ (b) に格納する。このアドレッシングモードはディスプレースメントが負の値を持った時、いわゆるプリデクリメント機能を有する。

【0057】データのプリフェッチ (b) は、プロセッサからデータキャッシュへのリクエストが無いときに、モディファイアドレスによりデータキャッシュユニットをアクセスする。データキャッシュユニットがヒットすればプリフェッチは完了するがミスしたらメモリアクセスを行う。

【0058】続いてプリフェッチ方式の詳細な説明を行う。図16はプリフェッチユニット105で、プリデクリメント用演算器1401、命令プリフェッチ用演算器1402、プリフェッチ用制御回路1403、データプリフェッチ用アドレスバッファ1405、命令プリフェッチ用アドレスバッファ1406、セクタ1407より構成する。

【0059】命令プリフェッチの動作を説明する。信号129でキャッシュミスしたことを報告されると、ミスした命令アドレス情報(物理アドレス)が128を介してプリフェッチユニットへ出力される。命令プリフェッチ用演算器1402は、命令アドレス情報+32バイトの演算を行う。この結果はミスしたブロックの次のブロックのアドレスである。この値は命令プリフェッチ用アドレスバッファ1406に格納される。その後、プリフェッチ制御回路1403よりプリフェッチリクエスト132とプリフェッチ用アドレス130を出力する。リクエスト132はメモリアクセス制御102内のマスク回路102aへ送られる。ここで、メモリ空間範囲指定用判定ユニット101の出力139により、もし、プロセッサが直接接続している自己分散メモリのアクセスであれば図12の1909はアサートされる。さらに、優先判定回路102cは信号1904をアサートさせプリフェッチユニットインタフェース制御1901にリクエストが受け付けられ、マスクされなかったことを知らせる。プリフェッチユニットインタフェース制御1901は、信号1904とメモリ空間範囲指定用判定ユニット101の記憶比較ユニット403~405からの信号139によってプリフェッチユニット105へ応答する信号を作成する。131aは応答信号であり、リクエストが受け付けられたときアサートする。131bは付属情報でありマスクされたか否かを応答するためのものである。この131a, 131bを受け取ったプリフェッチユニ

ット105は、リクエストが受け付けられたこと、そのリクエストに対してマスクされなかった事を知り、さらに、信号129を介して命令キャッシュユニット106に知らせる。

【0060】リクエスト1909がアサートされると、バス起動回路及びバスアービタ102dは通常とおり信号139によりリクエスト145を出力する。これによって、自己分散メモリをアクセスし内部データバス141を通して図5のデータ格納バッファ及びプリフェッチバッファ1205へ格納する。同時に、制御信号は146または148よりメモリアクセス制御102へ報告され、これを受けて102f, 131を伝わりプリフェッチユニット105へ知らされる。さらに、信号129を介して命令キャッシュユニット106に知らせる。

【0061】一方、マスク回路102a内で、メモリ空間範囲指定用判定ユニット101の出力139により、他の分散メモリやI/Oアクセスであることが認識されれば、図10の1909はネゲートされる。ネゲートされると、この部分でリクエストが終了し、転送バスへのプリフェッチが行われないことを意味する。

【0062】さらに、優先判定回路102cは信号1904をアサートさせプリフェッチユニットインタフェース制御1901にリクエストが受け付けられ、マスクされたことを知らせる。プリフェッチユニットインタフェース制御1901は、信号1904とメモリ空間範囲指定用判定ユニット101からの信号139によってプリフェッチユニット105へ応答する信号を作成する。131aは応答信号であり、リクエストが受け付けられたときアサートする。131bは付属情報でありマスクされたか否かを応答するためのものである。この131a, 131bを受け取ったプリフェッチユニット105は、リクエストが受け付けられたこと、そのリクエストに対してマスクされた事を知り、さらに、信号129を介して命令キャッシュユニット106に知らせる。

【0063】つまり、命令プリフェッチ要求は、プリフェッチユニット105より出されるが、メモリアクセス制御102内で自己分散メモリ空間か否かを判定し、自己分散メモリ空間のときのみアクセスを許し、それ以外のアドレス空間へのアクセスは許さないという制御を行う。これによって、転送バスには予測されたプリフェッチによる起動がかからず、転送バスの使用率を下げる効果がある。

【0064】続いて、データのプリフェッチの動作を説明する。データのプリフェッチは、プリフェッチアドレスを出力するまでに2つの方法を持つ。

【0065】ポストインクリメント機能を有する命令「LOADPI」「STOREPI」は、演算ユニット108内の演算器1302の出力がモディファイアドレスであるため、図16で1411のバスを通りセクタ1407を介してデータプリフェッチ用アドレスバッ

ァ1405に格納する。

【0066】一方、プリデクリメント機能を有する命令「LOADPD」「STOREPD」は演算ユニット108内の演算器1302の出力がモディファイアドレスとならないため、演算結果+ディスプレイメントの計算を演算器1401で計算する。この結果はモディファイアドレスとなりセクタ1407を介してデータプリフェッチ用アドレスバッファ1405に格納する。

【0067】2つの方法により、データプリフェッチ用アドレスバッファ1405に格納された論理アドレスは、プリフェッチ用アドレス150としてデータキャッシュユニット107へ送られる。図6のアドレス格納バッファ1004で受け、1023を通して論理アドレスラッチ1001に格納される。これ以降の動作はデータキャッシュユニット107の動作と同様である。つまり、データキャッシュユニットがヒットした場合は信号129を介してプリフェッチユニット105へ完了の報告を行う。一方、ミスした場合、自己分散メモリ内のアクセスであればブロック転送(32バイト)であり、他の分散メモリ、I/Oであればサブブロック転送(8バイト)を行い信号129を介してプリフェッチユニット105へ完了の報告を行う。

【0068】つまり、データプリフェッチ動作は、命令プリフェッチ動作と異なり、データプリフェッチ要求はプリフェッチユニット105より出されるが、アドレスは論理アドレスであるため、始めにデータキャッシュユニット107で、そのデータが存在するか否かのチェックを行う。その後、ヒットであれば、プリフェッチは完了する。ミスであれば、データキャッシュユニットの動作と同様に、メモリアクセス制御102内で自己分散メモリ空間か否かを判定し自己分散空間のときのみブロック単位のアクセスを許し、それ以外のアドレス空間へのアクセスはサブブロック単位のアクセスを許す制御を行いプリフェッチは完了する。これによって予測されたプリフェッチによる転送バスの使用時間を短くする効果がある。

【0069】以上、本実施例のプロセッサ内部からの外部メモリへのアクセスについて述べたが、本実施例の改善について述べる。

【0070】分散メモリ容量固定のシステムは、上限、下限レジスタ401、402をハードウェアで固定(たとえばROMメモリ化)してもよい。

【0071】さらに、比較回路404は、常に内部アドレスバス140が上限値レジスタ402より小さいことを検出するようにすればアドレス設定は図13(b)のようにすることもできる。

【0072】さらに、分散メモリの増減単位を例えば2の16乗(64Kバイト)と制限すれば、上位16ビットの比較でよく、レジスタ401、402にセットする値は図13(c)のようになる。これによって、比較回

路403、404、下限、上限レジスタ401、402を全て16ビットにすることができ、小型化できる。さらに、図9に示すメモリ空間範囲指定用判定ユニット101の変形として上限値または下限値アドレス(特定のベースアドレス)とそのメモリ空間範囲をあらかじめ設定しておき、内部アドレスバスが上記設定された範囲指定内か否か検出してもよい。

【0073】なお、本実施例において、上記レジスタ402にX' EFFFFFFF、下限レジスタ401にX' 00000000という値を設定すると、シングルプロセッサ構成のシステムを容易に構築することができる。

【0074】さらに、本実施例において、上記レジスタ402にX' FFFFFFFF、下限レジスタ401にX' 00000000という値を設定すると、転送バスをまったく使用しないシングルプロセッサ構成のシステムを容易に構築することができる。

【0075】さらに、本実施例において、上記レジスタ402にX' EFFFFFFF、下限レジスタ401にX' EFFFFFFFという値を設定すると、アドレスX' EFFFFFFF(システムとして使用禁止アドレス)以外は転送バスのみ使用することになり、従来のマルチプロセッサ構成のシステムを容易に構築することができる。

【0076】さらに、一般に、転送バスをサブブロック単位でアクセスするとバス使用時間は短くなる。しかしながら、連続するアドレスをアクセスするようなアプリケーションによっては、サブブロック単位のアクセスを4回(8バイト×4)よりもブロック単位のアクセス(32バイト)の方が転送バスを使用する使用時間は短い。なぜなら、転送するまでの初期動作が必ず入るためである。このため、自己分散メモリへのアクセスをサブブロック単位、その他の分散メモリへのアクセスをブロック単位とするようにバス起動制御102を改良することもできる。

【0077】さらに、プリフェッチにおいて本実施例では命令とデータの制御を変えているが、命令とデータともにどちらかに合わせて同じようにすることも簡単な修正で可能であるし、本実施例の命令とデータの制御を入れ替えることも可能である。さて、本実施例に戻って、今までとは観点を替えて、プロセッサの外部からのプロセッサアクセス要求について述べる。プロセッサアクセスは2つに分けられる。

【0078】(1)他のプロセッサから本プロセッサの分散メモリへのアクセス。

【0079】(2)他のプロセッサから本プロセッサのデータキャッシュメモリへのキャッシュコヒーレントチェック。

【0080】(1)の動作を説明する。本プロセッサは転送バス111上のアドレスが自己分散メモリ範囲内か

どうかを932を通して転送バス制御回路901で常に監視しており(メモリ空間範囲指定判定ユニット内の判定回路と同じものを持っている)、範囲内の時、転送バス111から本プロセッサのアクセス要求がある認識し、転送バス制御回路901は、転送バスリクエスト148を図10のバス起動回路及びアービタ102dに出力して内部バス使用許可を求める。その後の動作はメモリバスと転送バスがデータを受渡しするように制御されメモリ転送を完了する。

【0081】続いて、(2)の動作を説明する。他のプロセッサから本プロセッサのデータキャッシュメモリへのキャッシュコヒーレントチェックは、図12でバス927, 142を介してデータキャッシュユニットにチェックすべきアドレスが送られる。図6内のデータキャッシュユニット107ではデータ用アドレスアレ1003でキャッシュ内に同じアドレスのものが更新されていないかどうかをチェックする。アドレス144の5-14ビットで1Kエントリのうちの1つを選択する。読みだされた物理アドレスページと上位13-32ビットのアドレスは比較器1014で比較される。その出力1037とキャッシュのコヒーレンス用情報をデータキャッシュ制御ユニット1007でチェックする。キャッシュミス、あるいは、ヒットしていても更新されていなければ動作は終了である。しかしながら、ヒットしてかつ更新されているときにはキャッシュのデータをメモリに書き戻す処理を行う必要がある。これによって常にデータの一致性を保証する。

【0082】データ用アドレスアレ1003は、2ポートの構成となっており、通常のプロセッサの動作とキャッシュコヒーレントチェックは同時に実行できる。さらに、プロセッサがキャッシュミスして、自己分散メモリをアクセスしている場合にも、どちらかの処理を止めることなく並行して実行できる。これは、マルチプロセッサを高性能化できる効果がある。

【0083】本実施例の改良としてメモリ構成、システム構成を変えたときのメモリ空間範囲指定用記憶、比較ユニットとの関係を図17, 図18, 図19, 図20を使って述べる。

【0084】図17は、いくつかの範囲に分けて実装した各分散メモリのメモリ空間を示したものである。このようにときのメモリ空間範囲指定用判定ユニット101は、図19に示すように下限値格納レジスタ401, 450, 上限値格納レジスタ402, 451, 内部アドレスバスがレジスタの値より大きいことを比較する比較器403, 452, 内部アドレスバスがレジスタの値より小さいことを比較する比較器404, 453, それぞれの領域を判定する判定回路454より構成される。プロセッサ1は、初期状態のときX' 00000000より始まる領域の上限、下限アドレスを401, 402にセットし、X' 70000000より始まる領域の上

限、下限アドレスを450, 451にセットする。これによって、分散メモリ1のメモリ空間範囲を認識できる。

【0085】図18は、クラスタによる階層をつけたときのシステム構成を示す。このシステムは、クラスター1(1650)、クラスター2(1651)より構成する。さらに、1601~1604はプロセッサ、1604~1607は分散メモリ、1608~1609はI/Oコントローラ、1610~1611はクラスタ間通信コントローラ、1612はネットワークで構成する。システムの各分散メモリ空間の構成は図3と同じとする。この時のメモリ空間範囲指定判定ユニットへの下限レジスタ、上限レジスタに図13(a)の値をセットする。

【0086】このシステムの動作は、図18において、プロセッサ1を基準に考えると、プロセッサ1がアドレスX' 00000100なるアドレスをアクセスする場合、プロセッサ1内部の制御により、プロセッサ1が直接接続している分散メモリ1へのアクセスであることを認識し、プロセッサ1はメモリバス1を起動して分散メモリ1(1604)へブロック単位のアクセスを行う。

【0087】一方、プロセッサ1がアドレスX' 4F000000なるアドレスをアクセスする場合、プロセッサ1内部の制御により、プロセッサ1が転送バスへのアクセスであることを認識し、プロセッサ1は転送バス1624へサブブロック単位に起動する。プロセッサ1, 2(1601, 1602)はクラスタ内転送バス12を常に監視しており、このケースでは、プロセッサ2が各プロセッサに直接接続している分散メモリへのアクセスか否かを調べる。本例では、プロセッサ2が直接接続している分散メモリ2へのアクセスであることを認識しデータを受け取る。プロセッサ2は転送バスより受け取ったデータより、分散メモリ2へのアクセスをするための準備としてプロセッサ2の内部で転送バスとメモリバス2を接続する。このような一連の操作により、プロセッサ1(1601)、クラスタ内転送バス12(1624)、プロセッサ2(1602)、メモリバス2、分散メモリ2(1605)と接続されアクセスすることが可能になる。

【0088】さらに、プロセッサ1がアドレスX' EF000000なるアドレスをアクセスする場合、プロセッサ1内部の制御により、プロセッサ1が転送バスへのアクセスであることを認識し、プロセッサ1はクラスタ内転送バス12へサブブロック単位に起動し、クラスタ間通信コントローラ12(1610)を通してネットワーク1612をアクセスする。その後、クラスタ間通信コントローラ34(1611)を通してクラスタ内転送バス34をアクセスする。プロセッサ3, 4(1603, 1604)はクラスタ内転送バス34を常に監視しており、このケースでは、プロセッサ

ー3~4が各プロセッサに直接接続している分散メモリへのアクセスか否かを調べる。本例では、プロセッサ4が直接接続している分散メモリ4へのアクセスであることを認識しデータを受け取る。プロセッサ4はクラスタ内転送バス34より受け取ったデータより、分散メモリ4へのアクセスをするための準備としてプロセッサ4の内部でクラスタ内転送バス34とメモリバス4を接続する。このような一連の操作により、プロセッサ1(1604)、クラスタ内転送バス12(1624)、クラスタ間通信コントローラ12(1610)、ネット網1612、クラスタ間通信コントローラ34(1611)、クラスタ内転送バス34(1625)、プロセッサ4(1604)、メモリバス4、分散メモリ4(1607)と接続されアクセスすることが可能になる。

【0089】図18のシステムの更なる改良点として、各プロセッサのメモリ空間範囲指定用判定ユニット101を図19のような2つの空間の指定範囲を判定できる構成とする。この時、各レジスタにセットする値を図21に示す。ここで注目すべき点は、レジスタ401、402は自己分散メモリの範囲であり、レジスタ450、451はクラスタ内の分散メモリの範囲であることである。こうすることによりさらに細かい制御が可能である。例えば、図16のシステムの動作は、プロセッサ1を基準に考えると、プロセッサ1が分散メモリ1(1604)へブロック単位(32バイト)のアクセスを、さらに、分散メモリ2(1605)に2つのサブブロック単位(16バイト)のアクセスを、さらに、分散メモリ3~4(1606~1607)にサブブロック単位(8バイト)のアクセスを起動することも可能である。具体的には、判定回路454により139aがアサートされればブロック転送、139aがネゲート、139bがアサートの時2つのサブブロック転送、139a、139bが共にネゲートの時サブブロック転送するようにバス起動制御102は制御する。つまり、自己分散メモリ、クラスタ内の分散メモリ、クラスタ間の分散メモリごとにブロック転送量を替えることが可能である。なお、この時、転送バスのプロトコルとしてアドレス、データ、プロトコル制御信号に加えて、アクセスする単位である転送量のサイズも制御信号と共に送れるようにする必要がある。

【0090】更に、先行プリフェッチによるメモリアクセスに対しては、プロセッサ1が分散メモリ1(1604)へブロック単位(32バイト)のアクセスを、さらに、分散メモリ2と分散メモリ3~4(1606~1607)にはアクセスしないようにすることも可能である。

【0091】本実施例は、1次キャッシュメモリだけを考えているが、図22は、プロセッサ201から204が2次キャッシュを持つ場合のシステム構成である。特

徴は、各プロセッサ201から204から専用のキャッシュバスを通して2次キャッシュメモリ(4101から4104)と接続する構成である。各プロセッサが2次キャッシュ構成を取ったときの本発明への展開は、同業者であれば容易類推可能であるため省略する。

【0092】さらに、図23は、図1の改良として、プロセッサ内部でマルチプロセッサ構成を取ったときのブロック図を示す。このプロセッサは、演算ユニット1、演算ユニット2を持ち、主要信号は4214、4221が命令アドレスバス、4213、4220がオペランドバス、4215、4222が命令キャッシュとの制御信号、4211、4224がデータアドレスバス、4210、4223がデータバス、4212、4225がデータキャッシュとの制御信号である。命令キャッシュユニット106、データキャッシュユニット107は2ポートのキャッシュメモリ構成を取り、演算ユニット4201と演算ユニット4202は内蔵キャッシュメモリを共有したマルチプロセッサである。このようなプロセッサ内部でマルチプロセッサ構成を取った場合も本発明を展開することが可能である。

【0093】

【発明の効果】本発明の第1の効果は、分散共有メモリ型のマルチプロセッサシステムにおいて、通信バスの占有する時間を短くすべく、バス要求回数の低減と1回当りの転送量の最適化をおこない、通信バスの使用率を下げ、マルチプロセッサ全体の処理能力を向上させることである。

【0094】さらに、本発明の第2の効果は、プロセッサシステムにおいて、最も重要なことはメモリアクセスすべきデータがあらかじめキャッシュメモリに存在していることであり、今後使われるであろうデータに対して先行してメモリアクセスを起こす先行プリフェッチを、転送バスのバスネックを起こさないようにしながら可能な限りおこない、マルチプロセッサ全体の処理能力を向上させるシステムを提供することにある。

【図面の簡単な説明】

【図1】本実施例のプロセッサのブロック図。

【図2】図1で示したプロセッサを使用した共有分散メモリ型のマルチプロセッサの構成。

【図3】図2の分散メモリのアドレス空間。

【図4】演算ユニット108の構成。

【図5】命令キャッシュユニット106の構成。

【図6】データキャッシュユニット107の構成。

【図7】パイプライン動作。

【図8】サブブロック有効ビットの内部情報。

【図9】メモリ空間範囲指定用判定ユニット101の構成。

【図10】メモリアクセス制御の構成。

【図11】メモリバス制御103の構成。

【図12】転送バス制御104の構成。

【図13】システム構成各下限、上限レジスタにセットするアドレス値。

【図14】命令、データのプリフェッチの概念図。

【図15】ロード命令、ストア命令の命令機能。

【図16】プリフェッチユニット105の構成。クラスタによる階層をつけたときのシステム構成。

【図17】複数の範囲に分けて実装したときの各分散メモリのメモリ空間。

【図18】クラスタによる階層をつけたときのシステム構成。

【図19】2つの空間の指定範囲を判定できるメモリ空間範囲指定用判定ユニット101の構成。

【図20】図19の構成の判定回路。

【図21】クラスタによる階層をつけたときのシステム構成各下限、上限レジスタにセットするアドレス値。

【図22】2階層キャッシュを持つシステム例。

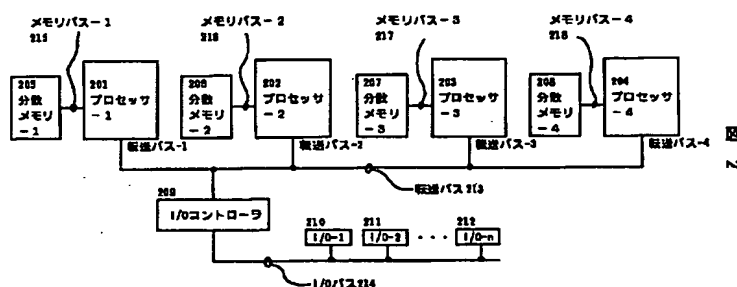
【図23】チップ内マルチプロセッサの例。

【符号の説明】

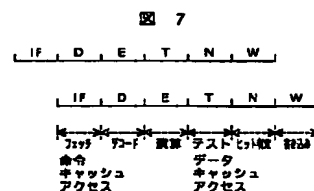
100…プロセッサ、101…メモリ空間範囲指定用判定ユニット、102…メモリアクセス制御、103…メモリバス制御、104…転送バス制御、105…プリフェッチユニット、106…命令キャッシュユニット、107…データキャッシュユニット、108…演算ユニッ

ト、110…メモリバス、111…転送バス、122…オペランドバス、123…命令アドレス、125…データバス、126…データアドレス、130…プリフェッチアドレス、131…リクエスト132に対する応答信号、132…プリフェッチリクエスト、133…命令アドレスリクエスト、134…リクエスト133に対応する応答信号、135…データキャッシュユニットから出力しているデータアドレスリクエスト、136…リクエスト135に対する応答信号、137…命令キャッシュユニットから出力している命令アドレス、139…メモリ空間範囲指定用記憶、比較ユニットから出力する範囲指定結果信号、140…内部アドレスバス、141…データキャッシュユニット107、命令キャッシュユニット106、メモリバス制御103、転送バス制御104を結ぶ内部データバス、145…バス起動制御から出力しているメモリバス起動リクエスト、147…転送バス起動リクエスト、201～204…図1で示したプロセッサ、205～208…分散メモリ、209…I/Oコントローラ、210～212…ディスプレイ、プリンタ、ディスク等のI/Oデバイス、213…転送バス、214…I/Oバス、215～218…各プロセッサ201から204が分散メモリ205から208までと接続するメモリバス。

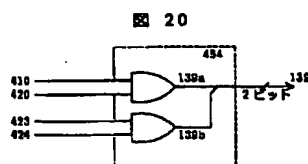
【図2】



【図7】

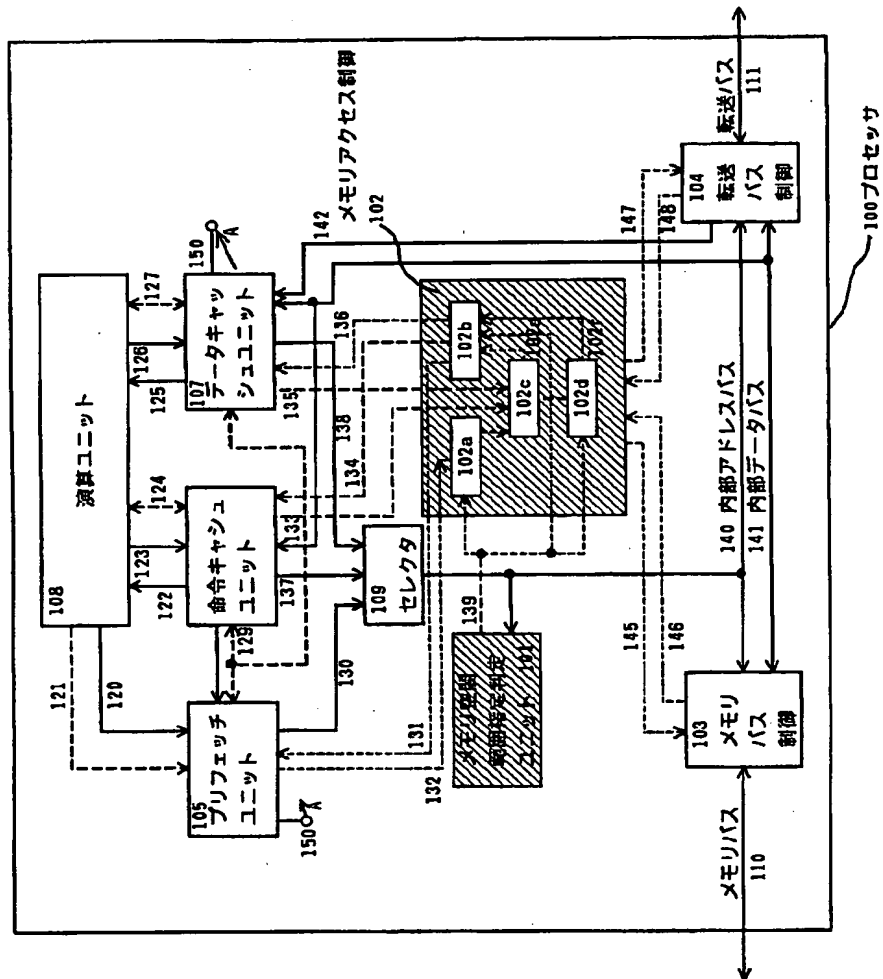


【図20】

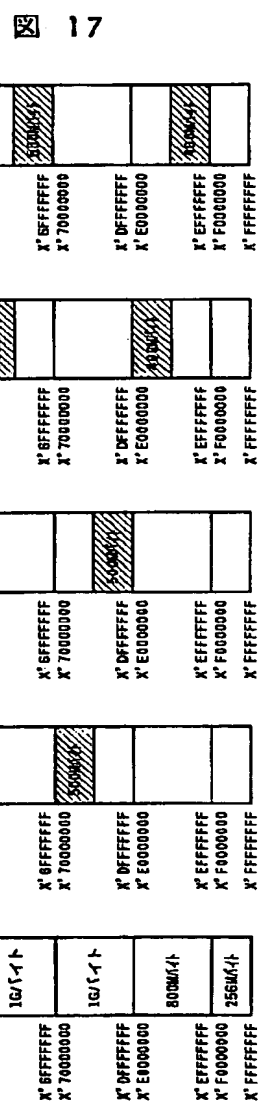


【図1】

図 1

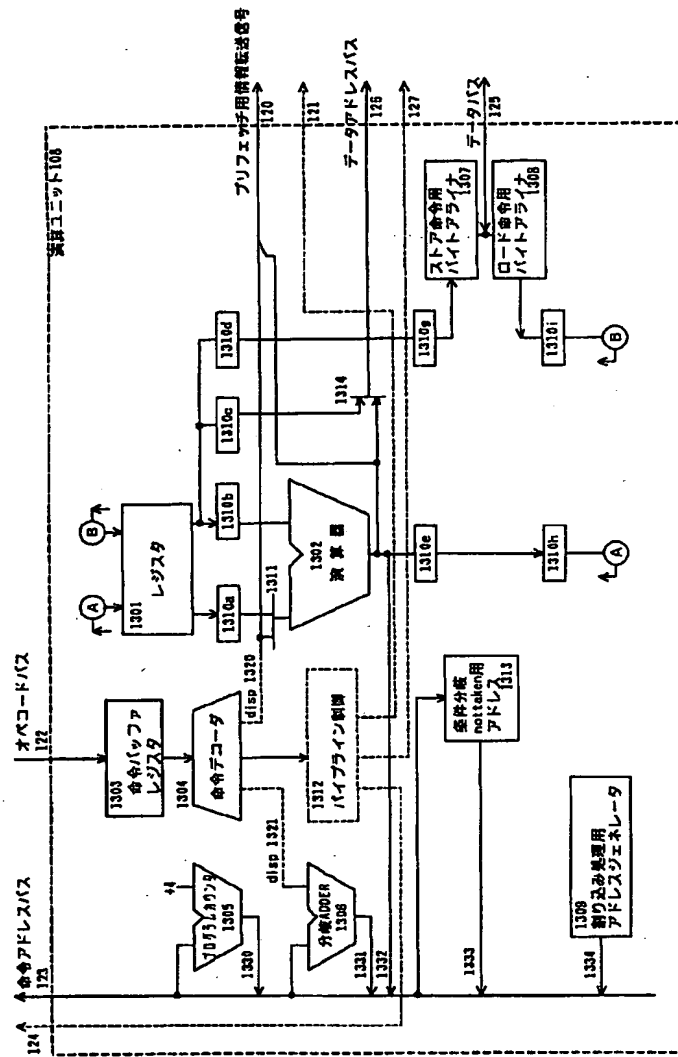


【图 1 7】



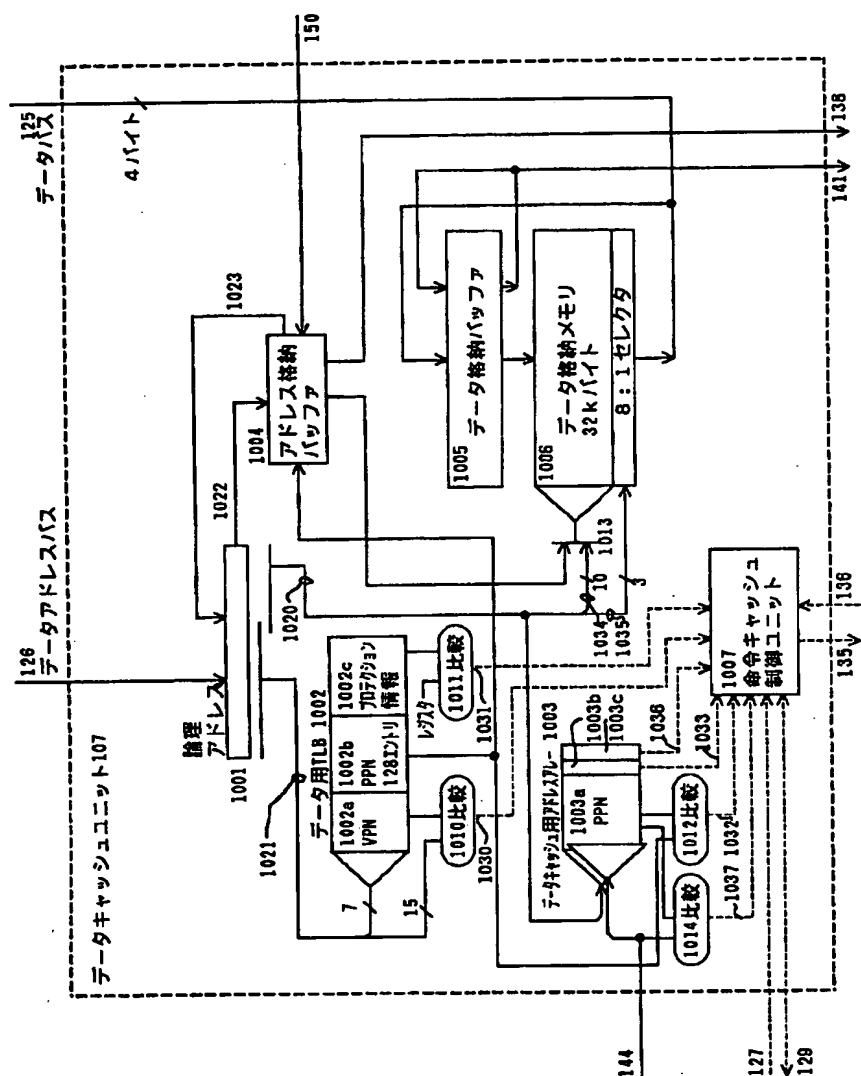
【図4】

図 4

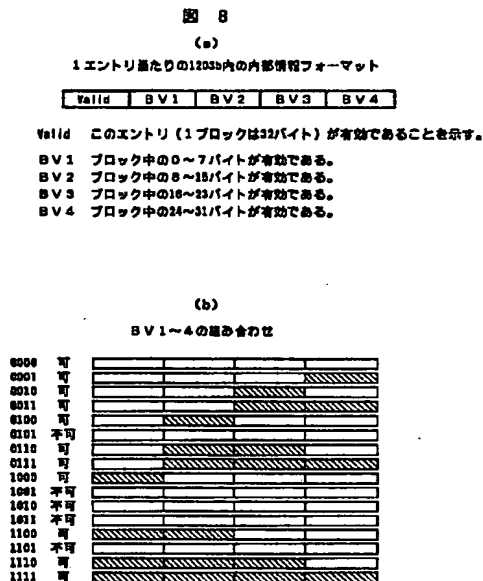


【图6】

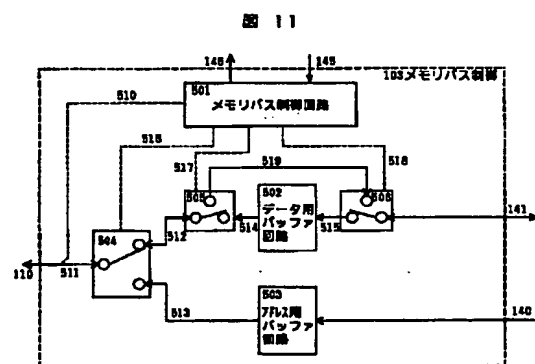
图 6



【図8】



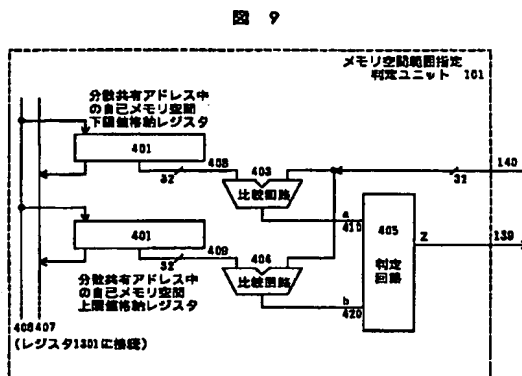
【図11】



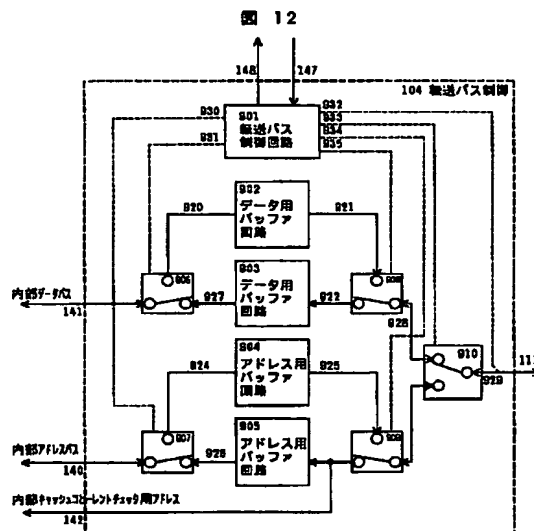
【図21】

	プロセッサ-1	プロセッサ-2	プロセッサ-3	プロセッサ-4
下位レジスタ 401	00000000	00000000	00000000	00000000
上位レジスタ 402	00000000	00000000	00000000	00000000
下位レジスタ 450	00000000	00000000	00000000	00000000
上位レジスタ 451	00000000	00000000	00000000	00000000

【図9】

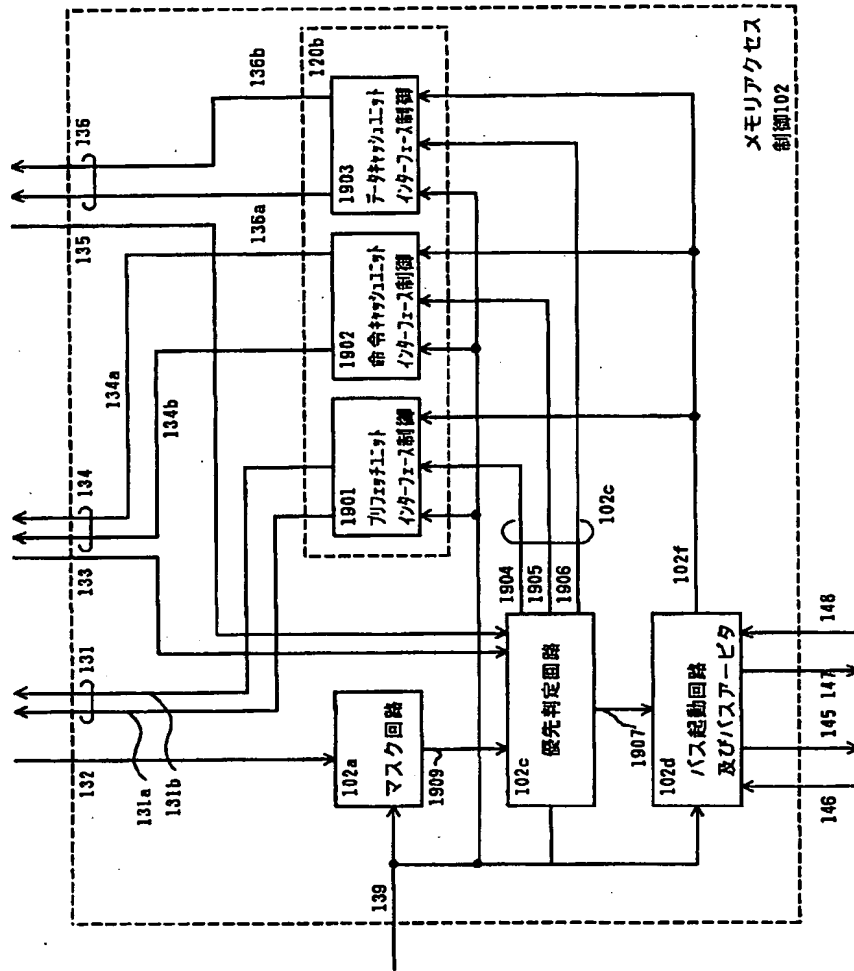


【図12】

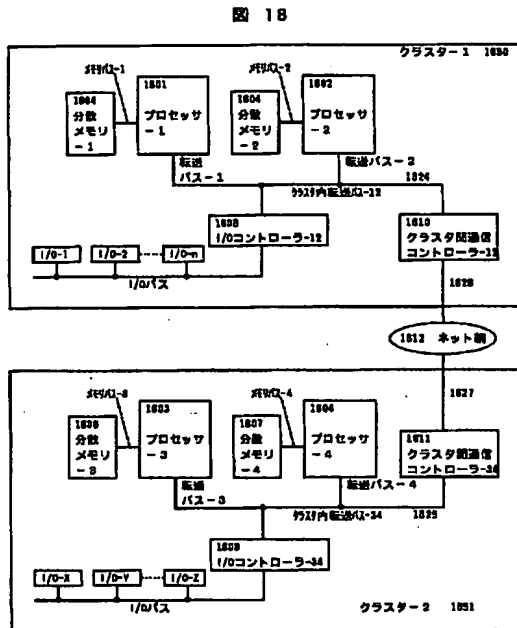


【図10】

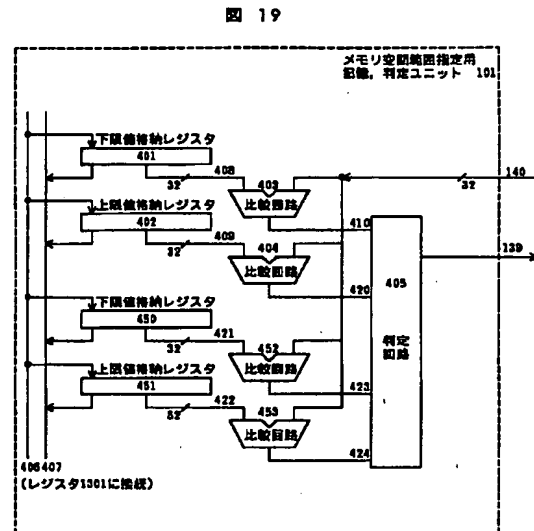
図 10



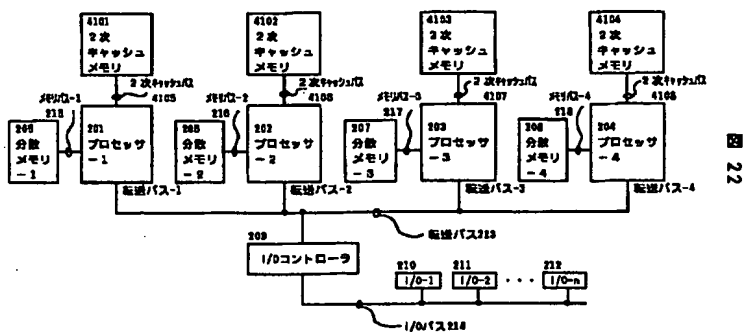
【図18】



【図19】



【図22】



【図23】

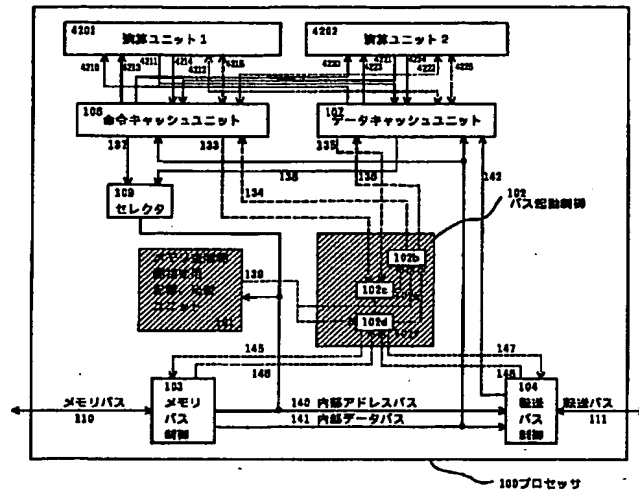


図 23

フロントページの続き

(72)発明者 森岡 道雄
 茨城県日立市大みか町七丁目1番1号 株
 式会社日立製作所日立研究所内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.